

ErasmusPlus
Applicazioni e Servizi Web

Filippo Venturini - 0001085749 {filippo.venturini8@studio.unibo.it}

Alberto Di Girolamo - 0001086465 {alberto.digirolamo2@studio.unibo.it}

Pietro Lelli - 0001085500 {pietro.lelli2@studio.unibo.it}

17 Luglio 2023

Contents

1	Introduzione	3
2	Requisiti	4
2.1	Requisiti utente	4
2.1.1	Studente	4
2.1.2	Admin	5
2.2	Requisiti funzionali	5
2.3	Requisiti non funzionali	5
3	Design	6
3.1	Metodologie di sviluppo	6
3.2	Target User Analysis	6
3.3	Mockups iniziali	9
3.4	Design architetturale	16
3.4.1	Architettura generale	16
3.4.2	Backend e Database	17
3.4.3	Frontend	17
3.5	Storyboard	18
4	Tecnologie	29
4.1	Socket.IO	29
4.2	Axios	30
4.3	Mongoose	30
4.4	Crypto-Js	30
4.5	GoogleMaps API	30
4.6	ApexChart	31
5	Codice	32
5.1	Utilizzo dell'API di GoogleMaps	32
5.2	Aggiornamento real-time con Socket-IO	33
6	Test	34
6.1	Euristiche di Nielsen	34
6.2	Test di usabilità	35

7 Deployment	36
7.1 Creazione Database	36
7.2 Installazione	36
7.3 Esecuzione	37
7.4 Utilizzo	37
8 Conclusioni	38

Chapter 1

Introduzione

Il progetto **ErasmusPlus** nasce dalle esigenze maturate da studenti che vogliono partecipare al progetto Erasmus tramite la piattaforma della propria università. L'idea è quella di realizzare una Web App che possa essere integrata nel normale sistema informatico di un'università (sostituendo applicativi datati come ad esempio AlmaRM di UniBo) e faciliti le procedure di candidatura e selezione sia lato studenti che lato amministratore.

In particolare per gli studenti interessati a far parte del progetto Erasmus, si vuole fornire una visione comprensiva di tutte le offerte disponibili su una mappa europea, in modo da permettere allo studente di orientarsi tra le varie possibilità tenendo conto della loro posizione geografica. Per gli studenti che prediligono una visione più compatta verrà inoltre messa a disposizione una pratica lista delle offerte disponibili, filtrabile a seconda delle esigenze.

ErasmusPlus offrirà la possibilità di visionare il dettaglio di un'offerta, che comprenda tutte le informazioni essenziali (come numero di posti, durata scambio ecc.) e le presenti con una grafica chiara e coerente. Per ogni offerta l'applicativo fornirà un insieme di statistiche tipicamente di alto interesse per gli studenti, come costo della vita, efficienza dei mezzi pubblici, qualità delle aule e altri ancora. Le statistiche verranno raccolte tramite la compilazione di questionari da parte di studenti che hanno concluso il viaggio Erasmus e possono fornire giudizi in base a quanto vissuto durante la propria esperienza. Lo studente avrà la possibilità di presentare facilmente le proprie candidature e un sistema di notifiche si occuperà di avvertirlo quando la sua offerta viene accettata o rifiutata dall'università.

Lato amministratore l'applicativo metterà a disposizione un metodo semplice ed agevole per aggiungere, modificare o eliminare le offerte Erasmus e ovviamente per gestire le candidature degli studenti. Anche l'amministratore avrà a disposizione un servizio di notifiche che gli permetterà di rimanere aggiornato ogni qualvolta un nuovo studente presenta una nuova candidatura.

Chapter 2

Requisiti

2.1 Requisiti utente

2.1.1 Studente

- Login con credenziali istituzionali
- Possibilità di eseguire il logout
- Visualizzazione dei dati personali
- Visualizzazione delle offerte alle quali ci si è candidati, compreso lo stato della candidatura
- Visualizzazione e ricerca offerte Erasmus tramite mappa
- Visualizzazione e ricerca offerte Erasmus tramite griglia
- Visualizzazione dettagli offerta (numero di posti disponibili, ambito di studio, durata scambio)
- Visualizzazione delle notifiche relative al proprio utente
- Visualizzazione statistiche relative all'università partner
- Possibilità di candidarsi per uno scambio
- Possibilità di inserire un offerta tra i preferiti
- Possibilità di compilazione di un questionario di gradimento al termine dello scambio

2.1.2 Admin

- Login con credenziali istituzionali
- Possibilità di eseguire il logout
- Visualizzazione di tutte le candidature
- Accettazione o rifiuto delle candidature in base ai posti disponibili per ogni scambio
- Visualizzazione delle notifiche relative al proprio utente
- Aggiunta di nuove offerte di scambio
- Modifica delle offerte di scambio già presenti
- Cancellazione delle offerte di scambio già presenti

2.2 Requisiti funzionali

- Gestione di due tipologie di utenti: amministratore e studente
- Gestione sistema di notifiche studente:
 - Notifica iscrizione ad una nuova offerta
 - Notifica cambiamento stato di una candidatura
 - Notifica invio questionario
 - Notifica aggiunta di una nuova offerta
- Gestione sistema di notifiche admin:
 - Notifica ricezione di una nuova candidatura
 - Notifica ricezione di un nuovo questionario
 - Notifica aggiunta di una nuova offerta
 - Notifica modifica e cancellazione di una offerta
- Gestione sistema di statistiche in base ai questionari di gradimento compilati dagli studenti al termine dell'esperienza

2.3 Requisiti non funzionali

- Il sistema deve essere immediato e facile da utilizzare
- Il sistema deve essere reattivo
- Il sistema deve risultare facilmente estendibile

Chapter 3

Design

3.1 Metodologie di sviluppo

Nel processo di realizzazione del progetto, è stata adottata una metodologia di design chiamata UCD (User Centered Design). La scelta di questa metodologia è stata determinata dal fatto che, fin dalle prime fasi di progettazione si è voluta dare priorità alla HCI (Human Computer Interaction), concentrando l'attenzione sulle necessità degli utenti e sull'obiettivo di garantire un'usabilità ottimale per migliorare l'esperienza dell'utente. Per quanto riguarda l'implementazione dell'applicazione, il focus si è posto sul dover soddisfare le esigenze e le richieste degli utenti. Gli User a cui si farà riferimento successivamente sono rappresentati attraverso una serie di profili chiamati "personas", che riflettono un possibile pubblico di riferimento dell'applicazione.

È stato deciso di adottare il principio KISS (Keep It Simple, Stupid) per creare interfacce che contengono solo gli elementi essenziali e che sono facilmente accessibili all'utente.

3.2 Target User Analysis

Il target di utenti è composto da studenti universitari che vogliono intraprendere il percorso Erasmus+ e dall'Università che gestisce le corrispondenti candidature.

Per effettuare tutte le procedure consone per partecipare a questo progetto è necessario fare uso di una piattaforma web che rappresenti un supporto sia per lo studente e sia per l'università di appartenenza. Il servizio offerto deve quindi permettere di compiere tutte le procedure necessarie per partecipare al programma.

Il target di utenza, rispetto ad una comune applicazione web, possiede una ristretta cerchia di utenza. Di conseguenza è necessario realizzare la piattaforma web avendo bene in mente le funzionalità che deve avere per permettere di

soddisfare pienamente i requisiti degli utenti. Di conseguenza vengono riportate delle analisi strategiche per la realizzazione dell'applicativo:

- Conoscenza delle procedure necessarie per adempiere il progetto Erasmus+, che comprendono la sequenza di step che uno studente deve effettuare e di conseguenza anche tutti i procedimenti che l'amministratore del sistema deve compiere per permettere una corretta fruizione del servizio
- Ottimizzazione dell'applicativo: il servizio deve essere intuitivo poiché gli studenti universitari che lo utilizzeranno, molto probabilmente non ne avevano mai fatto uso in precedenza. Sapersi muovere nelle varie schermate in modo intuitivo rende l'uso del servizio più appagante.
- L'aspettativa del target di utenza è quella di poter usufruire dell'applicativo come supporto per il proprio periodo di scambio all'estero, di conseguenza non deve presentare anomalie progettuali o funzionali.

Definiti i principali punti chiave da tenere in considerazione è possibile presentare un'analisi che racchiude gli aspetti che il Target User di riferimento necessita di avere.

Come già detto in precedenza il target è suddiviso in due categorie:

- Studenti universitari: si fa riferimento a tutti gli studenti che decidono di partecipare al progetto Erasmus+ e che devono quindi preoccuparsi di adempiere a tutte le procedure richieste: analizzare le possibili offerte visualizzandone le relative informazioni, presentare la propria candidatura e compilare un questionario al termine dell'esperienza
- Università di appartenenza: rappresentata come amministratore di sistema, deve poter gestire tutte le offerte presenti nell'applicativo, permettendo di modificarne le informazioni ed eventualmente di aggiungerne altre. Deve poter inoltre gestire le candidature effettuate dagli studenti accettandole o rifiutandole.

Definito lo User Target, vengono riportati diversi personas, ovvero rappresentazioni ipotetiche di utenti del servizio che devono utilizzare l'applicativo per assolvere determinati compiti.

Personas: Federico

Federico è uno studente universitario che ha appena deciso di partecipare al progetto Erasmus+ tramite la sua università.

Scenario d'uso: Federico vuole informarsi delle possibili offerte disponibili, controllando se sono compatibili alle sue esigenze come l'ambito di studio, periodo di scambio e mesi richiesti. Federico vuole salvarsi le offerte più interessanti e una volta effettuata la scelta, vuole potersi candidare.

1. Federico accede al sito effettuando il login con le sue credenziali istituzionali

2. Utilizza la mappa e la visualizzazione tabellare per esaminare le offerte disponibili
3. Tramite i filtri disponibili riesce a ottenere una selezione di offerte papabili
4. Visualizzando in dettaglio le offerte che gli interessano, inserisce nei preferiti quelle a cui pensa di mandare la candidatura
5. Dopo aver selezionato le offerte effettua le sue candidature

Personas: Chiara

Chiara è una studentessa che vuole partecipare al progetto Erasmus+.

Scenario d'uso: Chiara ha già effettuato ricerche in merito alle offerte che la sua università le propone, di alcune ha già presentato la candidatura. Vuole poter controllare lo stato delle sue richieste per sapere se è stata accettata oppure no.

1. Chiara accede al sito effettuando il login con le sue credenziali istituzionali
2. Essendosi già candidata, non vuole informarsi su altre offerte e così decide di visualizzare il suo profilo per controllare lo stato di accettazione
3. Eventualmente avrebbe potuto controllare la presenza di notifiche, poiché nel caso in cui lei fosse stata accettata o rifiutata, avrebbe ricevuto una notifica in merito

Personas: Gabriele

Gabriele è uno studente universitario che tempo fa aveva deciso di partecipare al progetto Erasmus+.

Scenario d'uso: Gabriele ha appena tornato in Italia concludendo il suo periodo di permanenza all'estero, vuole informare la sua università di questo riportando un suo riscontro sull'esperienza vissuta.

1. Gabriele accede al sito effettuando il login con le sue credenziali istituzionali
2. Clicca sul menu per poter poi visitare la pagina del suo profilo personale
3. Nella sezione del profilo visualizza tutte le offerte a cui si era candidato
4. Seleziona l'offerta a cui era stato accettato e viene così reindirizzato alla pagina di dettaglio dell'università
5. Clicca sul bottone *Termina*
6. Gabriele ha a disposizione un questionario anonimo che deve completare in merito all'esperienza appena vissuta

Personas: Amministratore

L'università che possiede l'applicativo ErasmusPlus ha la possibilità di gestire l'intero processo di sviluppo del progetto Erasmus.

Scenario d'uso: l'amministratore di sistema deve poter accettare le candidature per le relative offerte.

1. L'amministratore accede al sito utilizzando le sue credenziali
2. La sua visione dell'applicativo web è differente da quella visualizzata dallo studente perché ha obiettivi diversi
3. Si dirige nel menu e seleziona la voce *Gestione candidature*
4. In questa pagina ha a disposizione la lista di tutti gli studenti che si sono candidati per le offerte disponibili
5. Utilizzando i bottoni messi a disposizione dal sito web accetta o rifiuta le candidature presenti. Il sito sostiene l'amministratore ad accettare il numero corretto di studenti in base ai posti disponibili per ogni singola offerta

3.3 Mockups iniziali

Per la realizzazione dell'applicazione, per quanto riguarda l'interfaccia utente, si è cercato di mantenere uno stile grafico semplice ed intuitivo, per consentire all'utente di muoversi tra le varie pagine disponibili e tra le varie informazioni in modo veloce e intuitivo.

Inizialmente in fase di design si è fatto uso di mockup per ottenere una prima versione grafica dell'applicativo e per poter mostrare un'insieme di funzionalità che si riteneva che l'applicativo dovesse avere; grazie a queste interfacce si sono potuti ottenere vari feedback da parte di utenti.

I mockup sono stati realizzati utilizzando il tool Figma.

In seguito i vari mockup realizzati in fase di progettazione:

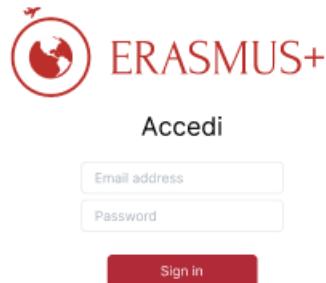


Figure 3.1: Login page

Il sito presenta all'apertura, come prima schermata, la pagina di login, attraverso la quale un utente può accedere al sistema utilizzando le proprie credenziali istituzionali.

Di seguito verranno mostrati i mockup che riguardano l'utente di tipo Studente:

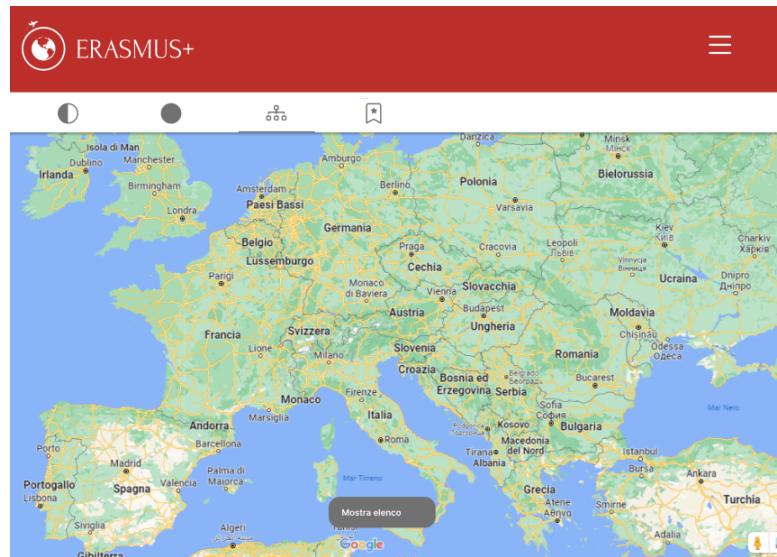


Figure 3.2: Homepage map

Dopo che la fase di autenticazione avrà avuto successo si verrà reindirizzati alla homepage principale in versione mappa.

Questa contiene dei filtri selezionabili, che andranno ad agire sulle offerte mostrate.

È presente inoltre un header comune a tutte le pagine che al suo interno contiene: un'immagine del logo, tramite la quale in qualsiasi momento si può

tornare alla homepage; un'icona che permette di visualizzare la lista delle notifiche e un'icona che permette di aprire un menù per navigare tra le varie pagine.

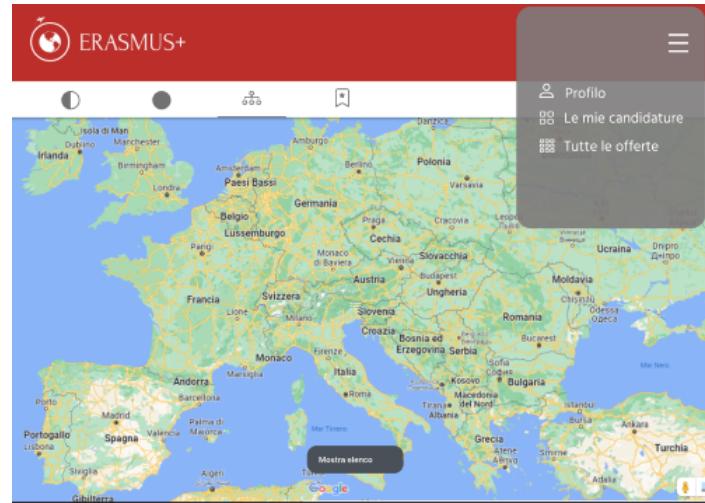


Figure 3.3: Menu homepage

Inoltre è presente un bottone che permette di cambiare la visione grafica delle offerte passando dalla visualizzazione a mappa ad una visualizzazione a lista, e viceversa.

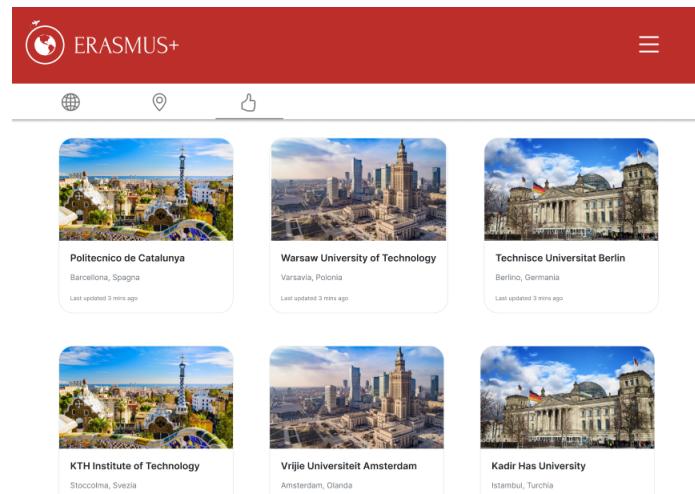


Figure 3.4: Homepage map

Questa versione della homepage consente all'utente di visualizzare in modo più rapido alcune delle informazioni più importanti relative alle offerte disponibili.

bili.



Figure 3.5: Offer page

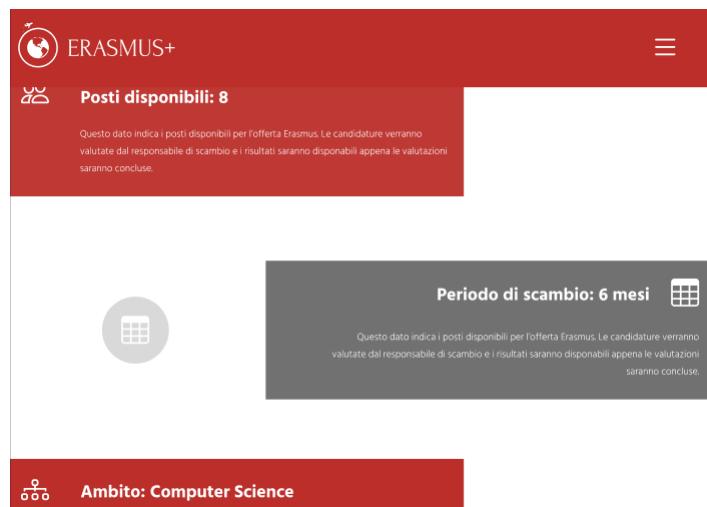
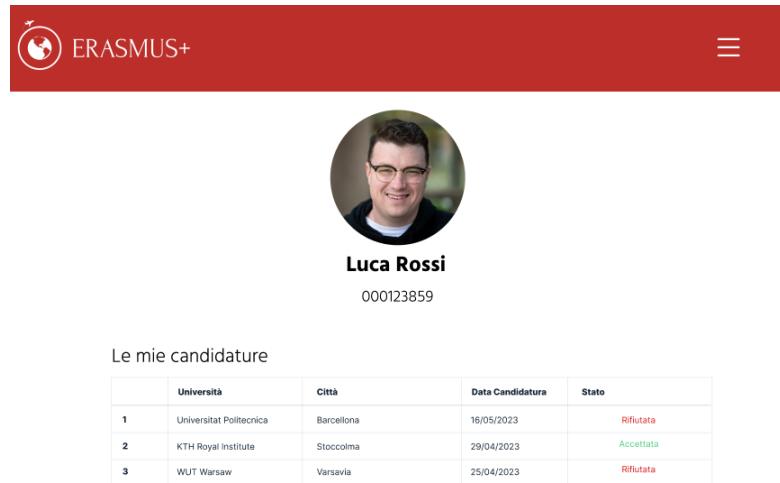


Figure 3.6: Offer page

Cliccando su una delle offerte presenti si verrà reindirizzati alla pagina di dettaglio dell'offerta. Saranno presenti immagini relative all'università e le informazioni utili, quali posti disponibili, durata dell'offerta, ambito di studio e verranno mostrate delle statistiche relative, calcolate sulla base di questionari

compilati dagli studenti.

Sarà inoltre possibile tramite un bottone candidarsi per l'offerta visualizzata.



The screenshot shows a user profile page for 'Luca Rossi'. At the top, there's a red header with the ERASMUS+ logo and a menu icon. Below the header is a circular profile picture of a smiling man with glasses. Underneath the picture, the name 'Luca Rossi' is displayed in bold black text, followed by the ID '000123859'. The main content area is titled 'Le mie candidature' (My applications) and contains a table with three rows of data:

	Università	Città	Data Candidatura	Stato
1	Universitat Politècnica	Barcellona	16/05/2023	Rifiutata
2	KTH Royal Institute	Stoccolma	29/04/2023	Accettata
3	WUT Warsaw	Varsavia	25/04/2023	Rifiutata

Figure 3.7: User page



The screenshot displays two pages from the user interface. The top part shows a 'Contatti' (Contacts) section with an email icon and the text 'pietro.letti@studio.unibo.it' and '3487634699'. To the right is a large envelope icon. The bottom part shows a 'Dati Personalni' (Personal Data) section with a person icon, containing the text 'Corso di Studio: Ingegneria e Scienze Informatiche' and 'Immatricolazione nell'a.a: 2021/2022'.

Figure 3.8: User page

Tramite il menù sarà inoltre possibile raggiungere la pagina relativa all'utente, qui verranno mostrate le sue informazioni e sarà presente una tabella riepilog-

ativa delle candidature alle quali ci si è iscritti, contenente anche lo stato della candidatura (in attesa, rifiutata o accettata).

Campus

	1	2	3	4	5
Servizi offerti (Mensa, biblioteca ecc.)	1	2	3	4	5
Aule	1	2	3	4	5
Laboratori	1	2	3	4	5
Spazi aperti	1	2	3	4	5

Città

	1	2	3	4	5
Clima	1	2	3	4	5
Costo della vita	1	2	3	4	5
Svago (Discoteche, Bar, Musei ecc.)	1	2	3	4	5
Mezzi pubblici	1	2	3	4	5

Figure 3.9: Survey page

Didattica

	1	2	3	4	5
Interesse ai corsi	1	2	3	4	5
Orari delle lezioni	1	2	3	4	5
Tutorati	1	2	3	4	5
Disponibilità dei docenti	1	2	3	4	5

Alloggi

	1	2	3	4	5
Costo	1	2	3	4	5
Disponibilità	1	2	3	4	5
Qualità	1	2	3	4	5
Vicinanza al campus	1	2	3	4	5

[Invia questionario](#)

Figure 3.10: Survey page

Dopo che lo studente avrà terminato la propria esperienza Erasmus, potrà dare una valutazione a quest'ultima tramite questo questionario, che riepiloga le informazioni più utili e interessanti che potranno servire agli studenti futuri. Una volta completato e inviato la pagina relativa all'offerta verrà di conseguenza modificata ricalcolando le statistiche e mostrando i grafici aggiornati.

Di seguito verranno mostrate le pagine dell'applicazione relative all'utente di tipo Admin:

The screenshot shows two sections of the application page. The top section is for 'Universitat Politècnica de Catalunya, Barcellona' with 3 available spots. It lists three students: Di Girolamo (16/05/2023), Lelli (29/04/2023), and Venturini (25/04/2023). The bottom section is for 'KTH Royal Institute, Stoccolma' with 3 available spots. It also lists the same three students. Each row has a 'Acceptation' column with a red 'X' or green checkmark.

Università	Studente	Data Candidatura	Accettazione
1 UPC Barcelona	Di Girolamo	16/05/2023	X ✓
2 UPC Barcelona	Lelli	29/04/2023	X ✓
3 UPC Barcelona	Venturini	25/04/2023	X ✓

Università	Studente	Data Candidatura	Accettazione
1 KTH Royal Institute	Di Girolamo	16/05/2023	X ✓
2 KTH Royal Institute	Lelli	29/04/2023	X ✓
3 KTH Royal Institute	Venturini	25/04/2023	X ✓

Figure 3.11: Application page

Questa pagina sarà raggiungibile dal menù e rappresenta la pagina di gestione delle candidature; l'admin potrà visualizzare tutte le candidature degli studenti divise per università, potrà decidere quindi, attraverso dei bottoni se accettarle o rifiutarle in base al numero di posti disponibili.



Figure 3.12: Offer page admin

La pagina di dettaglio della singola offerta verrà visualizzata dall'admin in questo modo. Saranno quindi presenti dei bottoni che consentiranno la cancellazione o la modifica dell'offerta.

In caso clicchi sul bottone modifica verrà visualizzata la pagina seguente, che permette di modificare tutte le informazioni e le immagini.

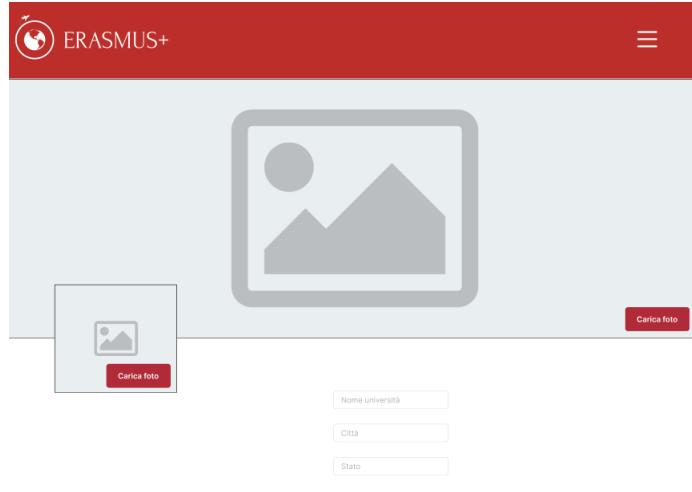


Figure 3.13: Update offer page

3.4 Design architetturale

Per quanto riguarda l'architettura del sistema, essa è stata progettata a partire dallo stack MEVN che comprende le seguenti tecnologie:

- MongoDB
- Express
- Vue
- Node.js

3.4.1 Architettura generale

MEVN è una delle possibili varianti dello stack MEAN e sostituisce per la parte di frontend Angular.js con Vue.js. In generale l'architettura si compone di tre parti principali: il frontend realizzato appunto con Vue.js, il backend realizzato con Node.js e Express e infine la parte di database per memorizzare i dati dell'applicazione realizzata con MongoDB. Internamente sono state utilizzate ulteriori tecnologie come Axios, Socket.IO e Mongoose per implementare la comunicazione e l'interoperabilità tra le componenti principali.

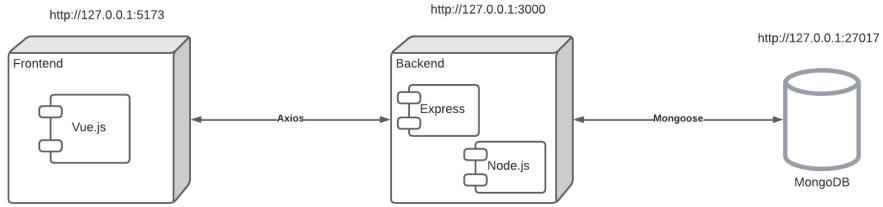


Figure 3.14: Architettura generale

3.4.2 Backend e Database

La parte di backend è organizzata a partire dai dati di cui l'applicazione necessita. In particolare sono stati identificati i seguenti gruppi di dati da cui è stata derivata la struttura del backend:

- **Users:** identificati come l'amministratore di sistema oppure come studenti, possiedono un'insieme di dati personali che riguardano l'ambito di studi, l'anagrafica universitaria e le informazioni necessarie per l'autenticazione.
- **Universities:** rappresentano un'offerta Erasmus in un'università estera, mantengono quindi tutti i dati core dell'applicativo che riguardano i dettagli dell'offerta, le informazioni riguardo all'università e le recensioni degli studenti.
- **Applications:** incapsulano le candidature presentate dagli studenti e gestite dall'amministratore.

Il design architettonico del backend è stato dunque organizzato implementando il pattern **MVC** e suddividendolo strutturalmente in **models**, **controllers** e **routes**. Di conseguenza ognuna delle categorie di dati precedentemente elencate possiede un proprio model, un proprio controller e un proprio router, che mette a disposizione tutte le RestAPI necessarie, mantenendo separate le varie parti del dominio applicativo.

3.4.3 Frontend

La parte di frontend è stata pensata con una suddivisione che favorisca la manutenibilità ed estendibilità del codice. In particolare si è progettato il sistema identificando per ogni pagina presente un **Component** di Vue.js. Al suo interno ogni componente contiene altri **Component** per le sottoparti più complesse, ognuna reattiva ai cambiamenti dei flussi di dati interni all'applicativo.

3.5 Storyboard

L'applicazione finale che si è realizzata non si discosta molto dai mockup realizzati in fase di progettazione, tuttavia sono state apportate delle modifiche tenendo in considerazione i feedback degli utenti.

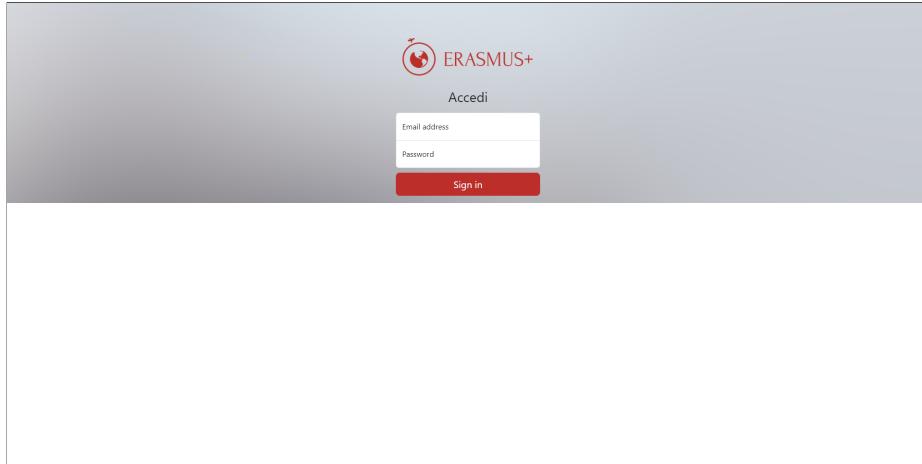


Figure 3.15: Login iniziale

La pagina iniziale dell'applicazione si presenta con questa schermata, che consente all'utente, sia che si tratti di uno studente, sia che si tratti di un amministratore, di accedere al sistema inserendo le proprie credenziali istituzionali. Nel caso in cui l'utente inserisca credenziali non valide gli verrà mostrato un messaggio di errore.

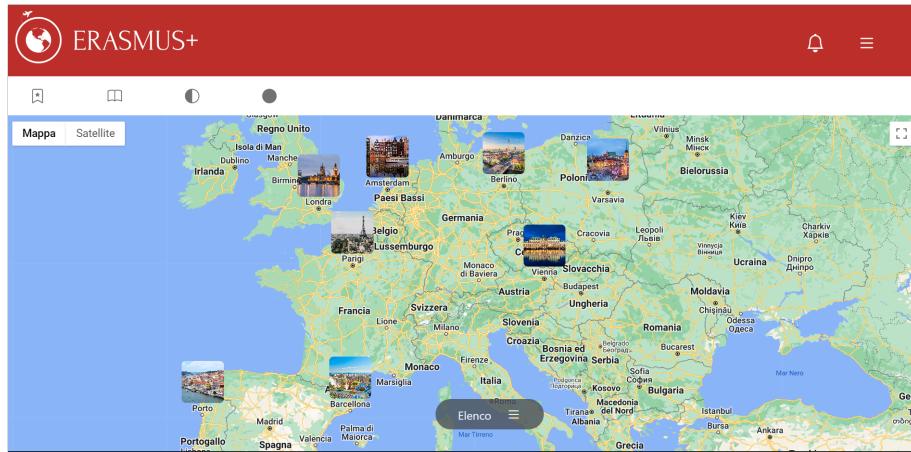


Figure 3.16: HomePage con la mappa

Una volta effettuato l'accesso l'utente sarà reindirizzato alla homepage, in versione di mappa. In questa pagina può visualizzare tutte le offerte disponibili disposte all'interno di una mappa, queste possono essere filtrate cliccando sugli appropriati filtri (preferiti, ambito di studio, durata offerta di 6 mesi o 12 mesi).

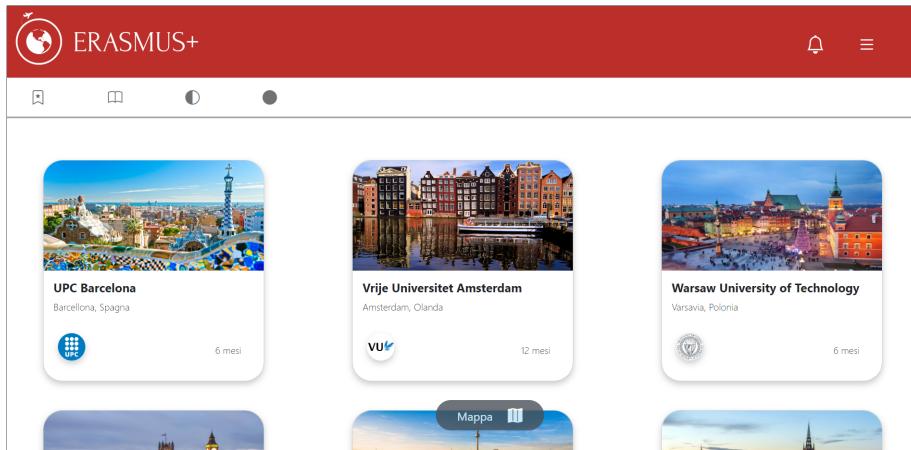


Figure 3.17: HomePage con la lista

Attraverso un bottone è possibile passare alla visualizzazione a griglia, che consente di individuare in modo più rapido alcune delle informazioni più importanti relative alle offerte disponibili.

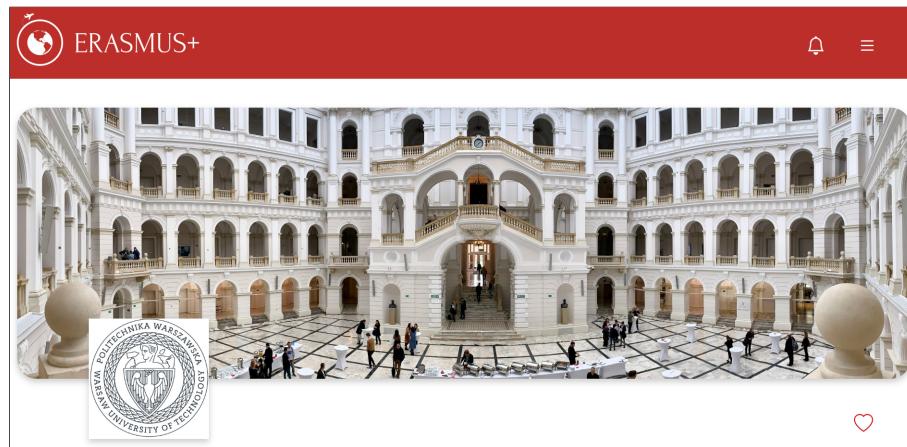


Figure 3.18: Intestazione dettaglio dell'università



Figure 3.19: Nome e locazione dell'università

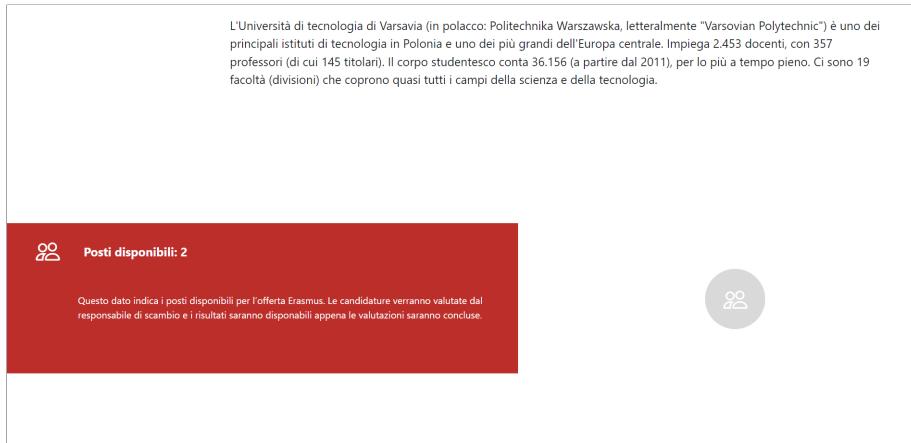


Figure 3.20: Posti disponibili dell'università

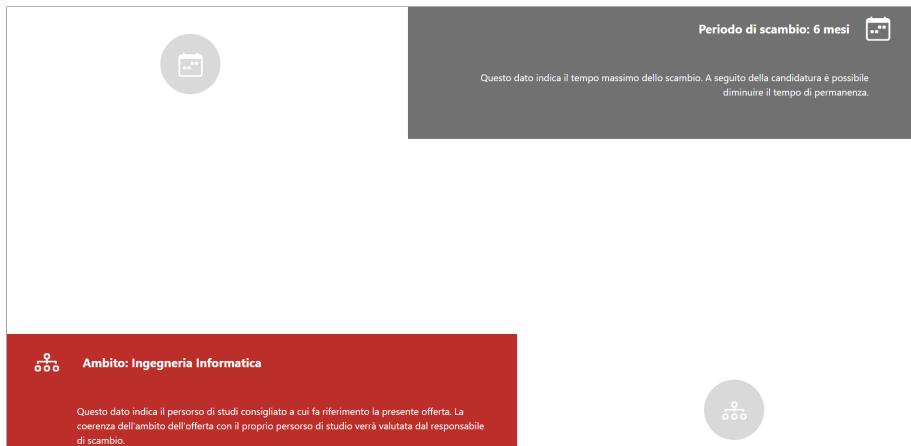


Figure 3.21: Durata scambio e ambito di studi

Cliccando su una delle offerte presenti si verrà reindirizzati alla pagina di dettaglio dell'offerta. Saranno presenti varie immagini relative all'università e alcune informazioni utili agli studenti, come ad esempio il numero di posti disponibili, la durata dell'offerta, l'ambito di studio e verranno mostrate delle statistiche relative, calcolate sulla base di questionari compilati dagli studenti. Lo studente potrà cliccare sull'icona del cuore per aggiungere l'offerta corrente ai preferiti.

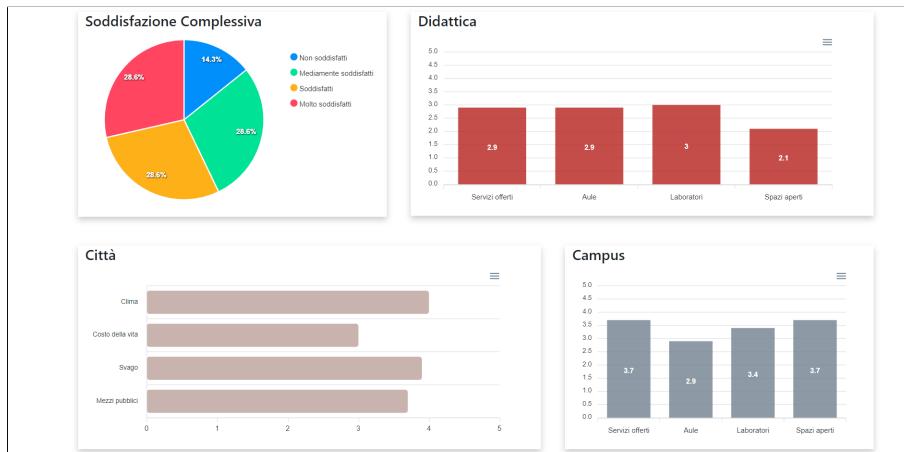


Figure 3.22: Statistiche offerta

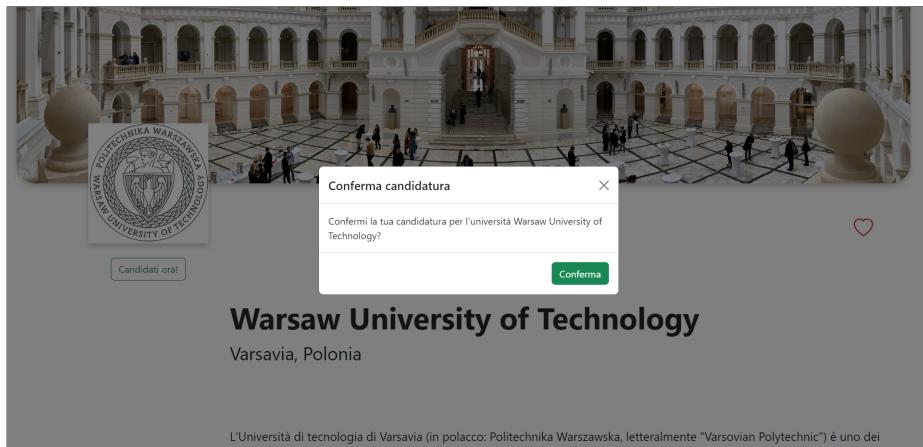


Figure 3.23: Presentazione della candidatura

Sarà inoltre possibile tramite un bottone candidarsi per l'offerta visualizzata dopo aver confermato la propria volontà di candidarsi attraverso la modal visualizzata nell'immagine appena mostrata.

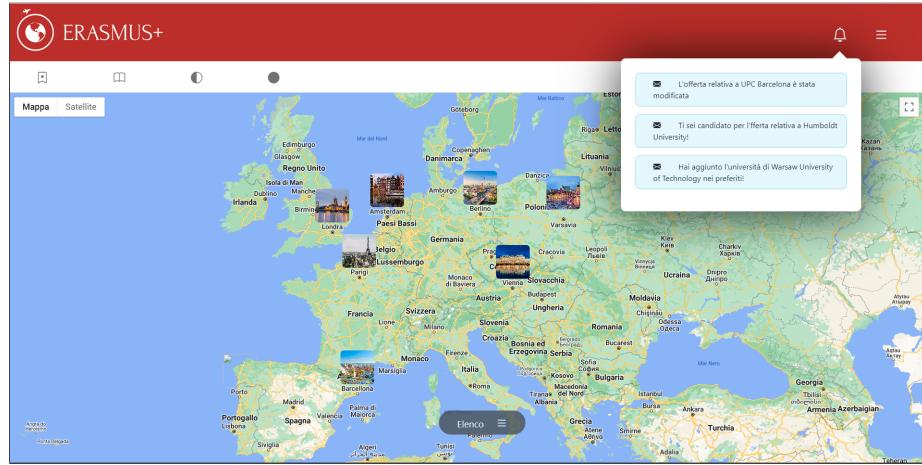


Figure 3.24: Visualizzazione delle notifiche

Da qualsiasi pagina, cliccando sull'apposita icona collocata nell'header, è possibile visualizzare la lista delle notifiche cliccabili relative al proprio utente.

Sempre dall'header è possibile aprire un menù che permette di navigare tra le principali pagine disponibili.



Figure 3.25: Pagina utente

Contatti

federico.minotti@studio.unibo.it
2443285539

Informazioni Personalni

Corso di Studio: Ingegneria Informatica
Immatricolazione nell'a.a. 2022/23

Figure 3.26: Pagina utente

Attraverso questa pagina l'utente può visualizzare i propri dati personali e i dati relativi alle proprie candidature, compreso lo stato di accettazione. Cliccando sul nome dell'università si verrà reindirizzati alla corrispondente pagina di dettaglio.

Campus

Servizi offerti (Mensa, biblioteca ecc.)	<input checked="" type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
Aule	<input type="radio"/> 1 <input checked="" type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
Laboratori	<input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input checked="" type="radio"/> 5
Spazi aperti	<input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5

Città

Clima	<input type="radio"/> 1 <input checked="" type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
Costo della vita	<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input checked="" type="radio"/> 5
Svago (Discoteche, Bar, Musei ecc.)	<input type="radio"/> 1 <input checked="" type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
Mezzi pubblici	<input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5

Didattica

Corsi pertinenti	<input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
Orazi delle lezioni	<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input checked="" type="radio"/> 4 <input type="radio"/> 5
Tutor	<input type="radio"/> 1 <input type="radio"/> 2 <input checked="" type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5
Disponibilità dei docenti	<input type="radio"/> 1 <input checked="" type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/> 5

Figure 3.27: Compilazione del questionario al termine dell’esperienza

Una volta terminata la propria esperienza presso un’università lo studente dovrà compilare un questionario inserendo votazioni da 1 a 5 a varie domande. Le risposte saranno poi analizzate e verranno mostrate nelle statistiche dell’offerta.

Di seguito verranno mostrate le pagine relative all’utente di tipo amministratore.

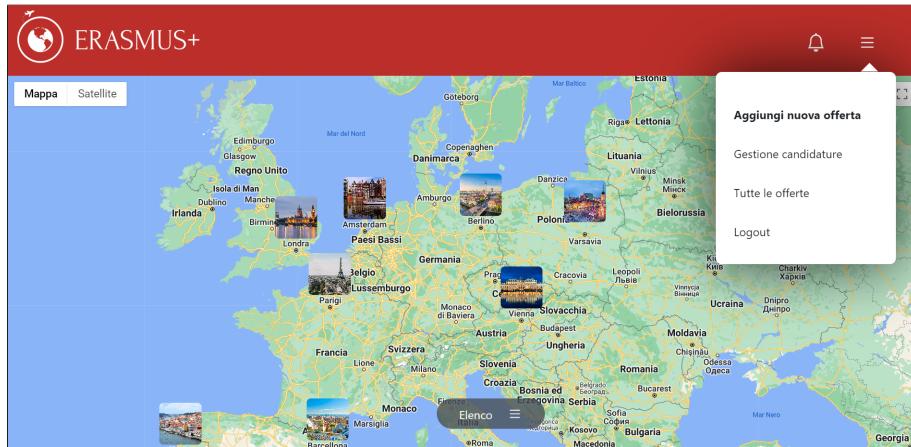


Figure 3.28: Menu dell’amministratore

Questo è come appare il menu dell’amministratore, da qui è possibile raggiungere la pagina per l’inserimento di una nuova offerta.



Nome dell'università

Città

Stato

Latitudine Longitude

Descrizione

Periodo: 6 mesi 12 mesi

Figure 3.29: Aggiunta di una nuova offerta

Descrizione

Periodo: 6 mesi 12 mesi

Logo:
 Nessun file selezionato

Immagine di copertina:
 Nessun file selezionato

Immagine per la miniatura:
 Nessun file selezionato

Ambito di studi:

Posti disponibili:

Figure 3.30: Dati richiesti per l'aggiunta di una nuova offerta

In queste immagini sono mostrati i dati richiesti all'amministratore per l'inserimento, comprese le foto che saranno poi mostrate agli utenti.



Figure 3.31: Dettaglio di un'offerta lato amministratore

Un'altra differenza di visualizzazione dell'amministratore rispetto allo studente è nella pagina di dettaglio di un'offerta, infatti qui gli verranno mostrati due bottoni che consentono di modificare o eliminare l'offerta corrente.

Figure 3.32: Modifica di un'offerta

Nel caso della modifica sarà aperta all'amministratore una nuova pagina con caselle di testo modificabili, contenenti inizialmente i dati correnti. Nel caso dell'eliminazione invece verrà visualizzata una modal per confermare la propria volontà di eliminare l'offerta.

Gestione candidature

UPC Barcelona

Posti disponibili: 2

	Università	Studente	Data candidatura	Accettazione
1	UPC Barcelona	Alberto Di Girolamo	16/05/2023	✓ ✗
1	UPC Barcelona	Pietro Lelli	16/03/2023	✓ ✗
1	UPC Barcelona	Martin Marcolini	11/02/2023	✓ ✗
1	UPC Barcelona	Nicolo Malucelli	11/05/2023	✓ ✗

Vrije Universiteit Amsterdam

Posti disponibili: 5

Figure 3.33: Pagina di gestione delle candidature

University of Oulu

Posti disponibili: 5

	Università	Studente	Data candidatura	Accettazione
1	University of Oulu	Pietro Lelli	10/01/2023	✓ ✗

University of P.Porto

Posti disponibili: 2

	Università	Studente	Data candidatura	Accettazione
1	University of P.Porto	Filippo Venturini	19/05/2023	✓ ✗

Sorbonne Université

Posti disponibili: 1

Non sono presenti candidature per questa università

Figure 3.34: Offerte senza candidature

Infine, un'altra pagina raggiungibile esclusivamente dall'amministratore è la pagina di gestione delle candidature. Qui è possibile visualizzare tutte le candidature degli studenti divise per università.

Attraverso dei buttoni l'utente amministratore potrà rifiutare le candidature o accettarle, fino al limite di posti disponibili per ogni offerta. Nel caso in cui venga accettata una candidatura di uno studente tutte le altre sue candidature risulteranno automaticamente rifiutate.

Chapter 4

Tecnologie

In questo capitolo vengono descritte nel dettaglio le tecnologie utilizzate che non appartengono allo stack tecnologico di base, ma che si sono resse indispensabili nella realizzazione dell'applicativo.

4.1 Socket.IO

Socket.IO è una libreria Javascript che fornisce API per l'implementazioni di real-time web applications e permette quindi una comunicazione bidirezionale in tempo reale tra client e server [6]. Tipicamente è composta da due parti, una viene gestita lato server e l'altra lato client, entrambe registrano delle socket che comunicano utilizzando differenti topic per gestire le varie interazioni. In ErasmusPlus Socket.IO si è resa fondamentale per la realizzazione del sistema di gestione delle candidature. Lato amministratore risulta indispensabile agevolare la meccanica di accettazione delle candidature, per questo tramite Socket.IO è possibile per visionare immediatamente il cambio di stato delle candidature in seguito ad un'accettazione o ad un rifiuto (non è necessario instaurare una comunicazione HTTP per richiedere aggiornamenti, ma vengono ricevuti in automatico). Con Socket.IO è stata quindi notevolmente incrementata la responsivness del sistema, in una fase critica che spesso risulta onerosa e complessa da parte dei coordinatori Erasmus.

4.2 Axios

Axios è una libreria promise-based HTTP per JavaScript [2]. Essa risulta uno strumento fondamentale per la comunicazione client-server in un'applicazione reattiva, in quanto permette di effettuare tutte tipologie di richieste necessarie per le operazioni CRUD (get, post, put, delete ecc.) e di ottenere per ognuna una promise. In questo modo, utilizzando il paradigma delle computazioni asincrone, è possibile registrare delle callback che specificano una routine da eseguire solamente dopo il completamento della richiesta. Di conseguenza, anche contenendo molte interazioni client-server tramite HTTP la web app mantiene un comportamento reattivo e consistente.

4.3 Mongoose

Mongoose è una libreria Javascript che permette di interfacciare applicativi Node.js con MongoDB [5]. In particolare questa libreria ci ha permesso di modellare secondo un certo schema i dati presenti nel nostro applicativo, riguardanti studenti, candidature e offerte. Questo agevola notevolmente l'implementazione della logica applicativa e fornisce delle linee guida che permettono di interagire con un database documentale come MongoDB in modo solido e coerente. Mongoose fornisce infatti la possibilità di strutturare le query relative a tutte le principali attività CRUD, per creare, leggere e modificare i dati memorizzati.

4.4 Crypto-Js

Crypto-Js è una libreria JavaScript che permette di utilizzare algoritmi crittografici [3]. Nell'applicazione ErasmusPlus si è fatto uso di questa libreria per salvare le password degli utenti in modo più sicuro, utilizzando l'algoritmo SHA-256. Allo stesso modo, al momento del login, le password salvate sul database vengono decriptate utilizzando lo stesso algoritmo e confrontate con quelle inserite dall'utente.

4.5 GoogleMaps API

La parte centrale di ErasmusPlus è costituita dalla mappa presente nella homepage che permette ad uno studente di visualizzare tutte le offerte disponibili. Facendo parte del core dell'applicazione, si è scelto di utilizzare le API di GoogleMaps fornite da Google, in quanto presentano un forte supporto che risulta facilmente integrabile in un applicativo Vue [4]. Google fornisce un MapLoader che permette di caricare la mappa tramite una API Key con una chiamata asincrona, mantenendo quindi intatta la semantica reattiva dell'applicazione. Queste API mettono inoltre a disposizione la possibilità di personalizzare i Marker posizionati sulla mappa, in questo modo è stato possibile rendere la grafica più chiara e adatta allo scopo.

4.6 ApexChart

ApexChart è una libreria JavaScript che permette di integrare nel frontend di una web app grafici di vario tipo come: istogrammi, grafici a torta, grafici a barre e tanti altri [1]. In ErasmusPlus una parte fondamentale è identificata dalla presenza di statistiche che riguardano le offerte universitarie ed è indispensabile organizzarle e renderle interpretabili dagli utenti in modo chiaro e completo. Perciò facendo affidamento su questa libreria, nella pagina di dettaglio delle offerte sono stati mostrati tutti i dati necessari, tramite varie tipologie di grafici.

Chapter 5

Codice

5.1 Utilizzo dell'API di GoogleMaps

Come già anticipato, per implementare la mappa nella HomePage che contiene tutte le offerte disponibili sono state utilizzate le API fornite da Google.

```
<GoogleMap api-key="***" class="map" :center="center" :zoom="5">
  <template v-for="university in this.universities">
    <CustomMarker v-if="this.checkUniversity(university)" :options="{
      position: {lat: university.latitude, lng:
        university.longitude}, anchorPoint: 'BOTTOM_CENTER'}">
      <RouterLink class="nav-link" :to="{path:
        '/universitydetail/' + university._id}">
        <div style="text-align: center" class="marker">
          
        </div>
      </RouterLink>
    </CustomMarker>
  </template>
</GoogleMap>
```

Listing 5.1: Implementazione della mappa con GoogleMaps

In particolare, una volta importate correttamente le dipendenze necessarie e ottenuta una API key per poter usufruire del servizio, è stato utilizzato il component **GoogleMap**. Questo permette di settare vari parametri generali della mappa, come il centro (individua la posizione geografica in cui sarà centrata la mappa all'apertura) e il livello di zoom desiderato. Al suo interno sono stati definiti più **CustomMarker**, uno per ogni offerta presente. Un **CustomMarker** è un component che individua un marker sulla mappa e permette quindi di settarne le coordinate, ma in più è personalizzabile e infatti in questo caso è stata inserita un'immagine al suo interno. Infine tramite l'utilizzo del **RouterLink**,

ogni marker è stato reso cliccabile e applica un reindirizzamento verso la pagina di dettaglio dell'offerta corrispondente.

5.2 Aggiornamento real-time con Socket-IO

Per implementare alcune features dell'applicativo mantenendo la massima responsività e reattività, si è reso necessario l'utilizzo di Socket.IO. La libreria fa uso di canali di comunicazione per i quali viene specificato un *topic*. In questo modo è possibile utilizzare diversi canali di comunicazione che corrispondono a diversi eventi che si verificano all'interno dell'applicativo. In particolare è stato configurato Socket.IO lato server per emettere sul topic "APPLICATIONS" l'insieme di tutte le candidature presenti nel database.

```
exports.sendUpdatedApplications = (data) =>{
    io.emit('APPLICATIONS', data);
}
```

Listing 5.2: Utilizzo di Socket.IO server side

Lato client è stato sufficiente registrare tramite la socket, una callback sul topic application, che permette di aggiornare in modo responsive e senza necessità di effettuare una richiesta HTTP la pagina delle candidature dell'amministratore, ricevendo i dati dal server.

```
this.socket.on('APPLICATIONS', (data) => {
    this.applications = data;
})
```

Listing 5.3: Utilizzo di Socket.IO client side

Chapter 6

Test

Il sistema è stato testato su tutti i principali Browser di riferimento, quali Chrome, Edge, Firefox e Safari. In questo modo è stata verificata la stabilità dell'applicativo e la sua portabilità, rendendolo indipendente dalle configurazioni scelte dai vari utenti. Oltre a test effettuati lato client, anche le API progettate lato server sono state verificate e il loro risultato validato, per garantirne la correttezza e consistenza.

6.1 Euristiche di Nielsen

Essendo inoltre l'idea dell'applicativo nata da una carenza di usabilità dei sistemi già presenti per la gestione del programma Erasmus, si è reso fondamentale testare ErasmusPlus riferendosi alle euristiche di Nielsen. Di seguito vengono riportate nel dettaglio le varie verifiche:

1. **Visibilità dello stato del sistema:** I link dell'applicativo sono stati concepiti e organizzati in modo da rendere l'applicazione meno disper-siva possibile, ogni elemento cliccabile che prevede un reindirizzamento esplicita chiaramente la sua destinazione. Le funzionalità non disponibili sono state esplicitate chiaramente utilizzando apposite indicazioni.
2. **Corrispondenza tra il sistema e il mondo reale:** Tutte le icone inserite all'interno del sistema presentano una corrispondenza con i concetti conosciuti dagli utenti, come aggiunta ai preferiti, filtro per ambito di studi o durata del periodo di insegnamento.
3. **Controllo e libertà per l'utente:** Gli utenti hanno piene libertà di navigazione, tramite il menu è sempre possibile navigare tra le pagine del sito ed inoltre tramite l'header viene fornita la possibile scorciatoia per accedere alla HomePage cliccando sull'icona del sito.

4. **Consistenza e standard:** Tutte le icone a cui è associato un significato semantico sono le medesime, sia nelle pagine di dettaglio delle offerte che nei filtri della HomePage, inoltre lo stile grafico risulta coerente e aderente a determinati standard in tutte le pagine dell'applicazione.
5. **Prevenzione dell'errore:** La logica è stata concepita in modo da mantenere sempre l'utente in una posizione non ambigua, in cui non è possibile commettere errori o generare inconsistenze nel sistema, è inoltre sempre possibile tornare indietro.
6. **Riconoscimento più che ricordo:** Le interfacce sono progettate per essere riconosciute e permettere all'utente di orientarsi. Inoltre laddove necessarie brevi spiegazioni (come nei dettagli di un'offerta oppure nella pagina di compilazione del questionario) sono indicate con parole semplici ed efficaci.
7. **Flessibilità ed efficienza:** Le offerte risultano consultabili sia via mappa che via lista, per garantire flessibilità di utilizzo a seconda dell'esperienza e della confidenza degli utenti.
8. **Estetica e progettazione minimalista:** L'estetica è stata concepita con aspetti grafici minimalisti, che accentuano nella giusta misura il contenuto informativo importante e minimizzano allo stretto necessario le informazioni aggiuntive.
9. **Facilità di riconoscimento, diagnosi e risoluzione delle situazioni di errore:** Tutti gli errori sono riportati con descrizioni semplici e chiare del problema, inoltre per tutte le operazioni critiche come candidature, modifiche di offerte o cancellazioni è sempre richiesta una conferma all'utente.
10. **Documentazione:** Il sistema non necessita di particolare documentazione, essendo molto intuitivo, ma presenta comunque brevi e comprensibili descrizioni nelle componenti meno ovvie.

6.2 Test di usabilità

Oltre alle verifiche di usabilità tramite euristiche sono stati raccolti feedback iterativi da un gruppo di utenti. Sono stati assegnati i due differenti ruoli di amministratore e studente al gruppo di utenti ed è stato richiesto loro di svolgere determinati compiti senza suggerimenti o consigli.

- **Studenti:** Agli studenti è stato richiesto di navigare tra le offerte disponibili e reperire le informazioni di interesse per le varie offerte. Inoltre è stato chiesto di provare a presentare varie candidature e a compilare i questionari di gradimento.
- **Amministratori:** Per quanto riguarda gli amministratori, è stato richiesto di provare ad aggiungere, modificare ed eliminare un'offerta e infine di gestire le candidature presenti, accettandole o rifiutandole.

Chapter 7

Deployment

7.1 Creazione Database

La creazione del database risulta facilitata utilizzando MongoDB Compass:

1. Effettuare la connessione tramite MongoDB Compass sulla porta **27017**
2. Creare un nuovo database con nome **"erasmusPlusDB"**
3. Creare una nuova collezione con nome **"applications"**
4. Creare una nuova collezione con nome **"universities"**
5. Creare una nuova collezione con nome **"users"**
6. Entrare nella collezione **"applications"** e selezionare "ADD DATA"
7. Selezionare il file json a partire dalla root del progetto: /db/applications.json
8. Ripetere il procedimento analogo per **"universities"** e **"users"**

7.2 Installazione

1. Clonare il repository al seguente indirizzo:
<https://github.com/FilippoVenturini8/ErasmusPlus.git>
2. Tramite una Shell Bash spostarsi all'interno della cartella del progetto

3. Eseguire i seguenti comandi:

- (a) `cd client/`
- (b) `npm install`
- (c) `cd ..`
- (d) `cd server/`
- (e) `npm install`

Una volta eseguiti i comandi indicati l'applicativo è correttamente installato e pronto per essere eseguito.

7.3 Esecuzione

Per eseguire ErasmusPlus è necessario eseguire il backend e il frontend, per farlo è necessario eseguire i seguenti comandi partendo dalla root del progetto:

- 1. `cd server/`
- 2. `npm start`
- 3. `cd ..`
- 4. `cd client/`
- 5. `npm run dev`

Ora è possibile collegarsi ad ErasmusPlus tramite un browser all'indirizzo:

`http://localhost:5173/`

7.4 Utilizzo

Per utilizzare l'applicazione è possibile effettuare il login con i seguenti account:

Amministratore:

Email: admin@unibo.it **Password:** Admin

Studente:

Email:federico.minotti@studio.unibo.it **Password:**1234

Chapter 8

Conclusioni

Lo sviluppo del progetto ci ha permesso di migliorare le nostre capacità di lavoro in team e di sviluppo collaborativo.

Inoltre utilizzando lo stack tecnologico MEVN, ci siamo interfacciati con tecnologie tutt'ora largamente utilizzate in ambito lavorativo e ampiamente spendibili nella realizzazione di progetti di qualsiasi tipo.

Inizialmente il principale scoglio si è identificato nell'utilizzo di un database documentale come MongoDB, essendo la nostra prima esperienza con un approccio noSQL, ma al fronte di uno studio dettagliato, il team è riuscito ad apprendere tutti i concetti necessari.

Per quanto riguarda le tecnologie web, il framework front-end è risultato facilmente comprensibile e applicabile grazie ai laboratori svolti durante l'anno. Infine nella parte di backend si è applicata una struttura appresa e in seguito adattata durante le lezioni, ciò ci ha permesso di sviluppare un applicativo solido e con un architettura ampiamente scalabile.

In conclusione il progetto ha fortemente aumentato le competenze tecniche del team di sviluppo per quanto riguarda la realizzazione di applicativi web moderni.

Bibliography

- [1] ApexChart. <https://apexcharts.com/>.
- [2] Axios. <https://axios-http.com/>.
- [3] CryptoJS. <https://cryptojs.gitbook.io/docs/>.
- [4] Google Maps API. <https://developers.google.com/maps?hl=it>.
- [5] Mongoose. <https://mongoosejs.com/>.
- [6] Socket.IO. <https://socket.io/>.